IN THE SPECIFICATION

Please amend Page 16, Line 13 to Page 17, Line 3 as follows:

In response to the memory configurations and/or figures of merit determined by memory design and optimization program 280, code optimizer 240 may re-order and/or re-write selected portions of the compiled object code in order to achieve greater efficiencies and to better meet the constraints specified in user design criteria 270. For instance, code optimizer 240 and/or memory design and optimization program 280 can modify the object code to store one or more sparsely used variables into an address space that corresponds to a flash memory that is cheaper [[that]] than SRAM in terms of cell area, but slower in terms of write speed. Also, a variable name that counts errors and that is very infrequently used may be re-written by code optimizer program 240 so that successive writes to the variable can be stored in consecutive memory locations (as opposed to the same one as a standard compiler would [[so]] do).

Please amend Page 18, Line 24 to Page 19, Line 11 as follows:

FIGURE 3 depicts flow diagram 300, which illustrates the operation of exemplary processing system 100 according to one embodiment of the present invention. Initially, processing system 100 compiles a source code file prepared by the embedded system designer to produce an object code file (process step 305). Then, processing system 100 executes ISS program 220 to ~~simulated~~ simulate the execution of the compiled object code (process step 310). Processing system 100 also runs debugger program 235 in order to debug and edit the object code, if necessary, as it runs (process step 315). As it is executed, ISS program 220 monitors the memory accesses in simulated ASIC memory space 230 and gathers memory usage statistics (process step 320). Simultaneously, ISS program 220 creates or updates the memory access histograms in histogram file 250 (process step 325).